

Fitting Vector Autoregressive Models to Electroencephalography Signals

2102531 Term Project Report

NUNTANUT RAKSASRI
AKASIT AUPAIBOON
PAWARISSON JAITHAHAN
JITKOMUT SONGSIRI

Department of Electrical Engineering, Faculty of Engineering,
Chulalongkorn University, Bangkok, Thailand

December 9, 2015

Abstract

Neural network of human body is transmitted through electrical current. The electrical current induces an electrical field which can be measured, called Electroencephalography or EEG. Nowadays, EEG signals are applied to use for medical benefits such as identification symptoms. Mathematical model of the signal has to be known before using as an identifier. This study proposes Vector-Autoregressive model (VAR) for estimation multi-channel EEG signals. The parameters are estimated by maximum likelihood method. Model selection is chosen by Bayesian Information Criterion (BIC). Wald test is used to test how correlated between channels of EEG signals. The result shows the best fit of EEG model has 1.8395% of residue error and the pattern of model parameters are generated by the Wald test.

1 Introduction

Information transmission between neurons or nerve cells in a brain is transmitted by electrical charges which induces a magnetic field and creates an electrical field. Electroencephalography or EEG signals are measured from electrical fields around a head by electrodes. EEG signal has very small magnitudes because a skull and scalp which have great resistivity to an electrical field and internal and external noise. The signals are detected, amplified, filtered and converted into digital before being observed. EEG signals are classified into 5 bandwidths; delta (0.5-4 Hz), theta (4-7.5 Hz), alpha (8-13 Hz), beta (14-26 Hz) and gamma (>30 Hz). Each frequency of the signals can be used to identify the behavior of a brain [1].

EEG signal has been estimated and modelled with many models by several algorithms. In this study, we introduce Autoregressive (AR) model to estimate the signal, which is expressed as:

$$X(t) = \sum_{k=1}^p A_k X(t-k) + e(t)$$

where $X(t)$ is an EEG sample at time t , $e(t)$ is an error term and A_k is a matrix parameter of past data.

There are many studies that already had applied and estimated the EEG model into many forms such as Autoregressive (AR) model and Autoregressive-Moving-average (ARMA) model and make use of them.

Epilepsy is a brain disorder, which normal pattern of neuron activity is disturbed. EEG model was introduced to distinguish between epileptic patients and normal people [1]. The difference between these two groups was classified by the peak alpha frequency (PAF) of EEG signals. The EEG signal of epileptic patient had higher amplitude of absolute spectrum power than the normal person. The study estimated

the EEG signal in form of ARMA model which was the best for a model with sharp peaks and deep nulls frequency spectrum. Its parameters were estimated by Kalman smoother algorithm. The biomarker was used to identify an epileptic patient. The biomarker of any EEG bandwidth was calculated with range of frequency of the models, power spectral density and sampling frequency. The t-test was introduced to identify which channels or bandwidth was best for distinguish between two groups. The biomarker of the identifier should be significant different. The result showed that every band excepts Gamma2 had significant difference of spectrum power.

The study in [2] shows another application of EEG signal, which is used to identify the sleeping disorders in human. The sleeping can be separated into 2 sleeping states; Rapid eyes movement (REM) and Non-rapid eyes movement (NREM). The data which was used was from the standard system of sleep recording and stating criteria called R&K standard. The standard of classifying between these two stages was created by visual classification by specialists. EEG signals were estimated in a form of ARMA model, which their coefficients were estimated by Kalman filter modeling. Feature vectors were generated from ARMA parameters by K-mean segmental HMM (Hidden Markov Model). The classification of sleeping states came from an observation sequence of the feature vectors. The result of classification was coherent to the R&K standard. However, it was hard to identify between 2 sleeping states. A medical opinion is still required.

For an AR model, in the study [3] aims to establish the feature for identifying a person by his/her EEG signal. The study based on the hypothesis that EEG signals were related to individual genetic information. This study seeked to find a biometrics which were AR parameters to identify person. The AR parameters were estimated by Yule-Walker approach. The parameters were formed as a feature vectors and were classified by Competitive Neural Network. The study compared feature vectors from single and multiple channels estimation of EEG models and found that feature vector from multiple channels estimation had more correctly than single channel estimation.

2 Formulating VAR equation of EEG signals

To formulate VAR equation of EEG signals, sample data of each node in each period of a brain are used to estimate parameters in the equation. VAR model with p-order lag of EEG is shown as:

$$y(t) = c + A_1 y(t-1) + A_2 y(t-2) + \cdots + A_p y(t-p) + v(t) \quad (1)$$

where $y(t) \in \mathbb{R}^n$ of EEG signal at time t , $c \in \mathbb{R}^n$ and $A_i \in \mathbb{R}^{nxn}$ are matrices of parameters and $v(t)$ is a gaussian white noise with covariance matrix is $\Sigma \in \mathbb{R}^{nxn}$. The equation (1) can be written as:

$$y(t) = AH(t) + v(t) \quad (2)$$

where $H(t) = (1, y(t-1), y(t-2), \dots, y(t-p)) \in \mathbb{R}^{np+1}$, $A = [c \ A_1 \ A_2 \ \cdots \ A_p] \in \mathbb{R}^{nxnp+1}$, and $v(t) \sim \mathcal{N}(0, \Sigma)$

VAR equation of EEG signals is estimated by maximum likelihood estimation. The parameters of the equation have to be estimated by maximizing the probability density function for the given parameters which are A and Σ . The probability density function for the given A and Σ can be formulated from the equation (2) as:

$$\begin{aligned} v(t) &= y(t) - AH(t) \\ f_y(y(t)|A, \Sigma) &= f_v(y(t) - AH(t)) \end{aligned} \quad (3)$$

Suppose data of EEG signal at $t = 1$ to $t = p$ are known, data y at $t = p+1$ to $t = N$, which N is a number of final estimation time point, need to be estimated. Therefore, the parameters of VAR equation which use to estimate data y at $t = p+1$ to $t = N$ can be solved from the likelihood function of y .

$$\begin{aligned} f(y|\theta) &= f(y|A, \Sigma) = f(y(t), y(t+1), \dots, y(N)|A, \Sigma) \\ &= f(y(p+1), y(p+2), \dots, y(N)|A, \Sigma) \\ &= \frac{1}{(2\pi)^{\frac{(N-p)}{2}}} \cdot \frac{1}{|\Sigma|^{\frac{(N-p)}{2}}} \cdot \exp -\frac{1}{2} \sum_{t=p+1}^N (y(t) - AH(t))^T \Sigma^{-1} (y(t) - AH(t)). \end{aligned} \quad (4)$$

It is convenient to consider the log-likelihood function of (4)

$$\begin{aligned}\log(f(y|A, \Sigma)) = L(A, \Sigma) &= \frac{N-p}{2} \log |\Sigma^{-1}| - \frac{1}{2} \sum_{t=p+1}^N (y(t) - AH(t))^T \Sigma^{-1} (y(t) - AH(t)) \\ &= \frac{N-p}{2} \log |\Sigma^{-1}| - \frac{1}{2} \|L[Y - AH]\|_F^2\end{aligned}\quad (5)$$

where L is from $\Sigma^{-1} = L^T L$, $Y = [y(p+1) \ y(p+2) \ \dots \ y(N)] \in \mathbb{R}^{nxN-p}$,

$$\text{and } H = \begin{bmatrix} 1 & 1 & \dots & 1 \\ y(p) & y(p+1) & \dots & y(N-1) \\ y(p-1) & y(p) & \dots & y(N-2) \\ \vdots & \vdots & & \vdots \\ y(1) & y(2) & \dots & y(N-p) \end{bmatrix} \in \mathbb{R}^{np+1 \times N-p}.$$

The parameter \hat{A} is solved by minimizing the term $\|L[Y - AH]\|_F^2$ to maximize the probability density function by differentiating this term with A , which the result is shown as:

$$\hat{A} = YH^T(HH^T)^{-1} \quad (6)$$

The parameter $\hat{\Sigma}$ is solved by differentiating the equation (5) with Σ^{-1} , which the result is shown as:

$$\hat{\Sigma} = \frac{1}{N-p} \sum_{t=p+1}^N (y(t) - \hat{A}H(t))(y(t) - \hat{A}H(t))^T \quad (7)$$

the estimation of VAR model of EEG signal can be written as:

$$\hat{y}(t) = \hat{A}H(t) \quad (8)$$

The AR model is achieved, next step is to choose lag order of AR model which is accurate and reliable. Bayesian Information Criterion (BIC) is introduced to assess the models with various order in term of BIC score.

BIC score is calculated from:

$$\text{BIC} = -2L + d \log N$$

where L is log-likelihood function of y , d is the number of effective parameters and N is the number of samples. The proper lag order is found if BIC score of its model is the lowest.

3 Result and Discussion

10 channels of focal x-signal EEG of an epilepsy patient are estimated with different p^{th} lag order and are compared to the real data. In this experiment, VAR models with lag order equal to 1-20 are chosen to observe how lag orders affect to quality of VAR models.

Figure 1 shows the relation between the residue mean error of model of 10 channels with 10240 samples which its lag order vary between 0-20. The figure show that wtih an increase of lag order, the model is tend to has less error. However, it is hard to say that which lag order has the least order. Plus, higher lag order is also affect complexity of calculation.

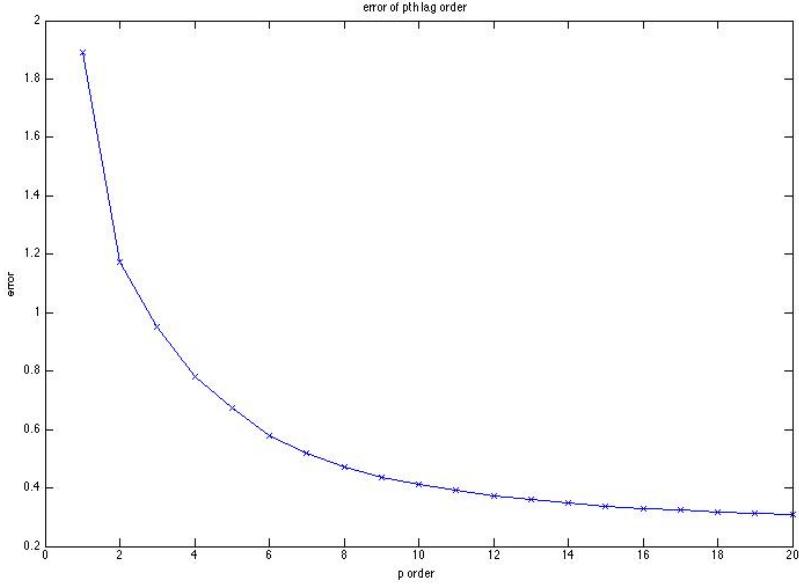


Figure 1: Graph Between Error Magnitude and pth lag order.

Figure 2 shows the relation between BIC score and lag order of models. At the lag order equals to 10 has the lowest BIC score which equals to 4.645×10^5 . However, other higher lag order such as 11, 12 have less residue error than 10th but they are slightly different.

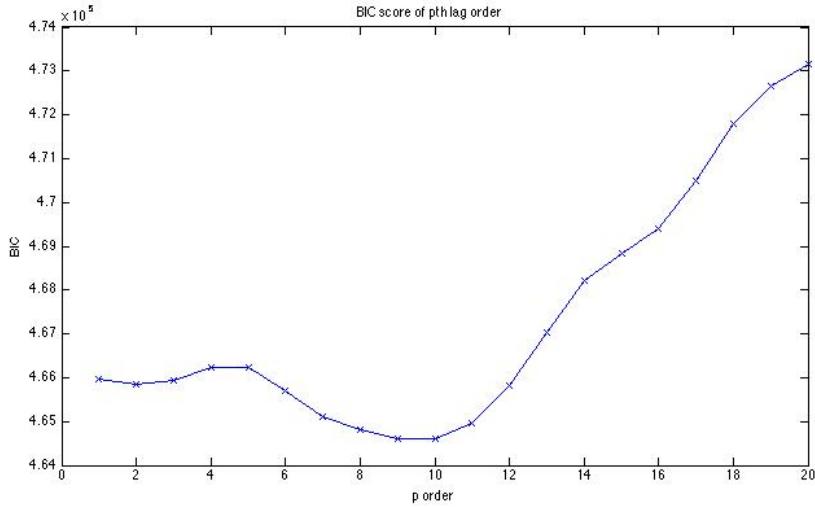


Figure 2: BIC score with p order.

Figure 3 shows the real EEG signal and the estimated model of lag order equals to 10. The real signal and the estimated model has 1.8395% of the residue error. AR model with lag order equals to 10 is the best fit for this EEG signals.

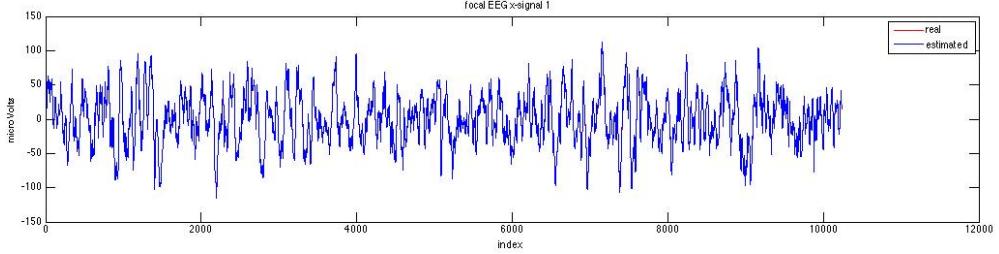


Figure 3: VAR Model with 10th-order.

4 Hypothesis Testing by the Wald Statistics

We are going to begin with assuming that an unknown parameter θ has been estimate by ML, that it belongs to a parameter space $\Theta \subseteq \mathbb{R}^p$, and we want to test the null hypothesis

$$H_0 : r(\theta) = 0$$

where $r : \mathbb{R}^p \rightarrow \mathbb{R}^m$ is a restriction function, with $m \leq p$.

Let $\hat{\theta}$ be the estimate of the parameter $\theta \in \mathbb{R}^p$ obtained by maximizing the log-likelihood function over the whole parameter space Θ :

$$\hat{\theta} = \arg \max_{\theta \in \Theta} L(\theta)$$

where $L(\theta)$ is the log-likelihood function.

From [5], the Wald test is based on the following test statistic:

$$W = r(\hat{\theta})^T [D_r(\hat{\theta}) \mathbf{Avar}(\hat{\theta}) D_r(\hat{\theta})^T]^{-1} r(\hat{\theta})$$

where $D_r(\hat{\theta})$ is Jacobian of r with respect to the entries of $\hat{\theta}$, and $\mathbf{Avar}(\hat{\theta})$ is a consistent estimate of the asymptotic covariance matrix of $\hat{\theta}$. Asymptotically, the test statistic has a Chi-square distribution, as stated by the following proposition.

Proposition : Under the null hypothesis that $r(\theta) = 0$, the Wald statistic W converges in distribution to Chi-square distribution with m degrees of freedom.

We can calculate $\mathbf{Avar}(\hat{\theta})$ by using the Fisher information matrix for θ contained in y (1 sample) which is defined as

$$\mathcal{I}(\theta) = \mathbf{E}[(\nabla_\theta L(\theta))(\nabla_\theta L(\theta))^T]$$

the expectation of the outer product of the gradient of the log-likelihood function. $\mathbf{Avar}(\hat{\theta})$ is defined as

$$\mathbf{Avar}(\hat{\theta}) = \frac{1}{N} \mathcal{I}(\theta)^{-1}$$

when $\mathcal{I}(\theta)$ is the Fisher information matrix for θ and N is the sample size. For the ML estimation, the asymptotic covariance matrix can be computed by

$$\hat{\mathbf{Avar}}(\hat{\theta}_{ml}) = \left[- \sum_{i=1}^N \nabla^2 \log f(y_i | \hat{\theta}) \right]^{-1}$$

The idea of the Wald test is if H_0 is true, $\hat{\theta}$ should satisfy $r(\hat{\theta})$ to be close to zero.
In the Wald test, the null hypothesis is rejected if

$$W > c$$

where c is a pre-determined critical value. The size of the test (significance level) can be its asymptotic value

$$\alpha = P(W > c) = 1 - P(W \leq c) \approx 1 - F(c)$$

where $F(c)$ is the distribution function of Chi-square random variable with m degrees of freedom. The critical value c can be chosen by

$$c = F^{-1}(1 - \alpha)$$

5 Performing hypothesis test on LS problem.

Consider, LS estimation

$$\hat{x} = \arg \min_x \|Ax - y\| = (A^T A)^{-1} A^T y$$

with the restrictions,

$$r_{11}x_1 + r_{12}x_2 + \dots + r_{1k}x_k = q_1$$

$$r_{21}x_1 + r_{22}x_2 + \dots + r_{2k}x_k = q_2$$

...

$$r_{m1}x_1 + r_{m2}x_2 + \dots + r_{mk}x_k = q_m$$

and can be written in matrix form,

$$Rx = q$$

for example, if one of the coefficients is zero, $x_i = 0$,

$$R = [0 \ 0 \ \dots \ 1 \ 0 \ \dots \ 0] \text{ and } q = 0.$$

or some coefficients is zero such as, $x_1 = x_3 = x_4 = 0$,

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \text{ and } q = 0.$$

The hypothesis implied by the restrictions is

$$H_0 : Rx = q,$$

$$H_1 : Rx \neq q.$$

where H_1 is called alternative hypothesis.

By Wald test, if H_0 is true, then $R\hat{x} - q$ should be close to zero. For the example of a single coefficient, if $H_0 : x_i = 0$ is correct, then \hat{x}_i should be close to zero.

Now consider $y = Ax + \mathcal{V}$ where \mathcal{V} is gaussian white noise $\sim \mathcal{N}(0, \sigma^2 I)$, we choose x to be zero in some entries. Then generate y and estimate x using LS method. After that we perform Wald test on \hat{x} and detect the zero entries of x .

By using Matlab, we choose $x \in \mathcal{R}^{100}$ (some coefficients is zero), $A \in \mathcal{R}^{120 \times 100}$, $\mathcal{V} \sim \mathcal{N}(0, 10|x|)$, and use $\alpha = 0.05$. Then generate $y = Ax + \mathcal{V}$ 10,000 data set and calculate two type of errors by averaging the value of them over the data set. The result is

Type I error = 0.0505

Type II error = 0.0008

when the two types of error is the error that we reject the true H_0 called type I error and the error that we accept the false H_0 called type II error.

Now consider a value of a significance level, variance of noise, and sample size that the value of the type II error depend on it. Repeating the Wald test to calculate the Type II error by the following situation and the result is shown in Figure 4,5, and 6 that is when increasing the significance level, and the sample size the type II error tend to be decreasing and when increasing the variance of noise the type II error tend to be increasing.

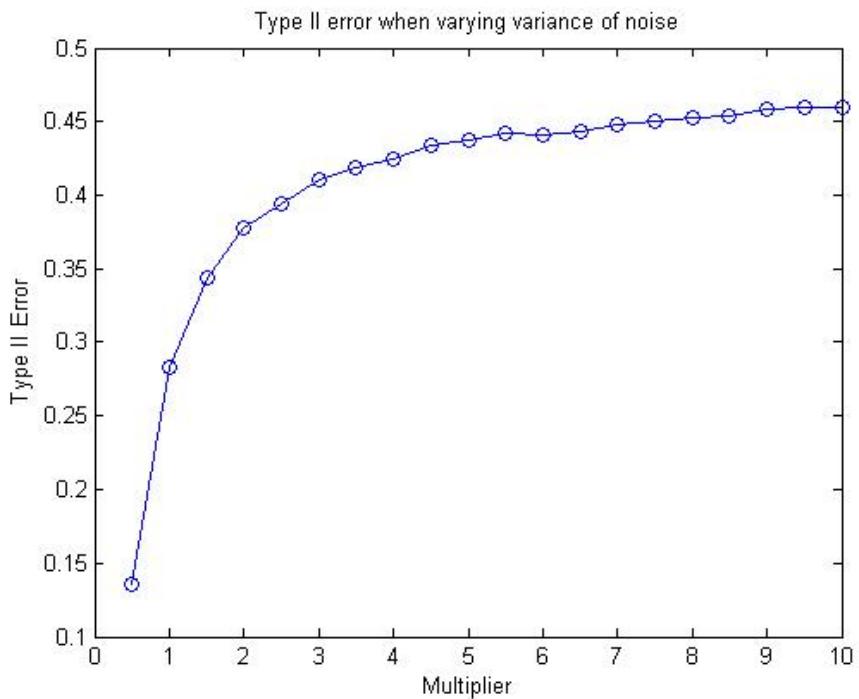


Figure 4: Type II error when varying the variance of noise

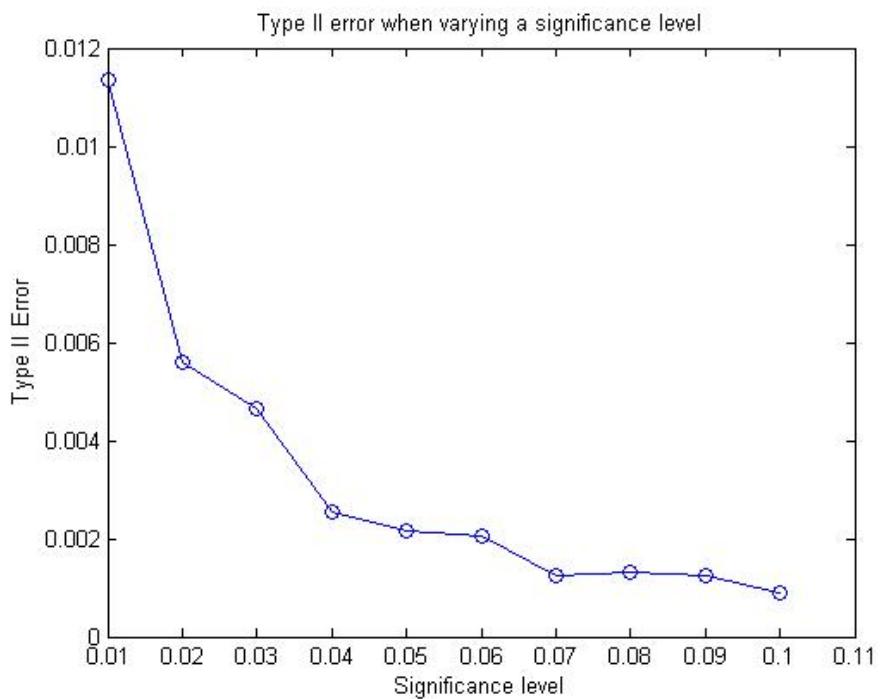


Figure 5: Type II error when varying the significance level

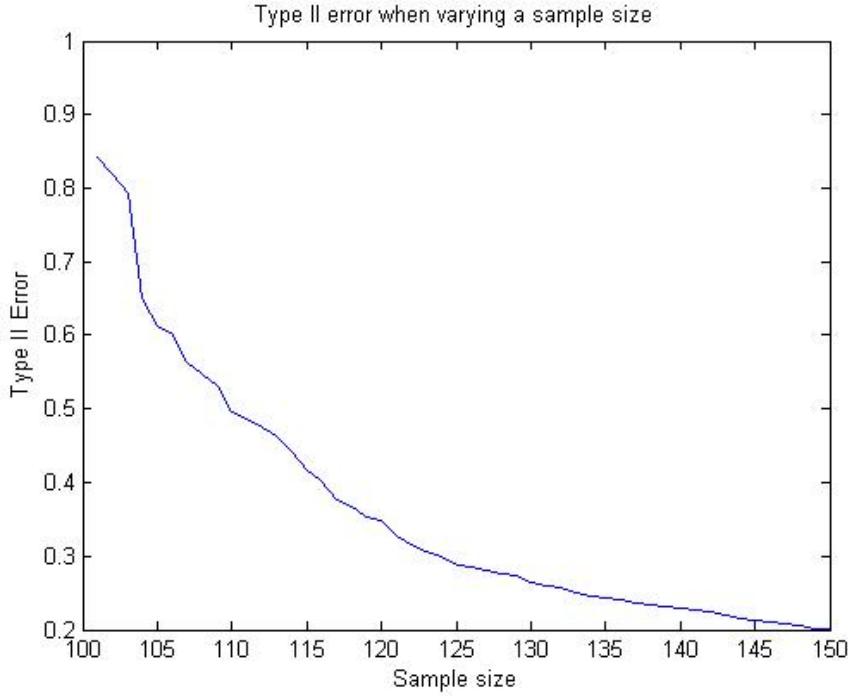


Figure 6: Type II error when varying the sample size

6 Performing hypothesis test on AR Model.

In this section we generate AR model : $y(t) = AH(t) + v(t)$ by choosing $y(t) \in \mathbb{R}^4$, $A \in \mathbb{R}^{4 \times 40}$, lag order $p = 10$ and significance level $\alpha = 0.05$ with sample size equal to 10,000 data points. Then compute the pattern of model parameter. The result is

$$\begin{aligned} \text{True model parameters pattern : } & \begin{bmatrix} X & X & 0 & X \\ X & X & X & X \\ X & 0 & X & 0 \\ X & 0 & X & X \end{bmatrix} \\ \text{and estimated model parameters pattern : } & \begin{bmatrix} X & X & 0 & X \\ X & X & X & X \\ X & 0 & X & 0 \\ X & 0 & X & X \end{bmatrix} \end{aligned}$$

when X represent to the non-zero value of the model parameter and 0 represent to the zero value of the model parameter.

From the result, y_3 is not has an affect on y_1 , y_2 and y_4 are not have an affect on y_3 , and y_2 is not has an affect on y_4 and the estimated model parameter pattern is the same as the true pattern.

7 Conclusion

From mathematics calculation and graph analysis, AR model is suitable to estimate multi-channels EEG signal. The best estimation AR model of this EEG signal, which has low error and sensible and reliable, is the model with lag order equals to 10 and has residue error of 1.8395% .In the performing hypothesis section, we can see the accuracy of the Wald test for analyzing a zero component in the model parameters when varying some values and for AR model we can analyze the pattern of model parameters to see the correlation between the time series data.

References

- [1] S. Sanei, J. A. Chambers, *EEG Signal Processing*. Centre of Digital Signal Processing, Cardiff University, 2007
- [2] T. A. Cassar, K. P. Camilleri, S. G. Fabri, M. Zervakis, S. Micheloyannis, *ARMA modeling for the diagnosis of controlled epileptic activity in young children*. Communications & Control and Signal Processing, 2008, p. 25-30
- [3] A. B. Rossow, E. O. T. Salles, K. F. Coco, *Automatic Sleep Staging Using a Single-Channel EEG Modeling by Kalman Filter and HMM*. Biosignals and Biorobotics Conference (BRC), 2011, p. 1-6
- [4] G. Mohammadi, P. Shoushtari, B. M. Ardekani, M. B. Shamsollahi, *Person identification by using AR model for EEG signals*. World Academy of Science, Engineering and Technology, 2006, vol. 11, p. 281-285
- [5] W. H. Greene, *Econometric Analysis*. Prentice Hall, 2008

Appendix

MATLAB Code

The Matlab code is consisted of the log-likelihood calculation, Plotting trade off curve, Plotting BIC score, Performing Wald test for LS problem, and Performing Wald test for AR model.

```
% import 10 EEG signals
eeg1 = importdata('Data_F_Ind0001.txt');
eeg2 = importdata('Data_F_Ind0002.txt');
eeg3 = importdata('Data_F_Ind0003.txt');
eeg4 = importdata('Data_F_Ind0004.txt');
eeg5 = importdata('Data_F_Ind0005.txt');
eeg6 = importdata('Data_F_Ind0006.txt');
eeg7 = importdata('Data_F_Ind0007.txt');
eeg8 = importdata('Data_F_Ind0008.txt');
eeg9 = importdata('Data_F_Ind0009.txt');
eeg10 = importdata('Data_F_Ind0010.txt');

data = [ eeg1(:,1)' ;
          eeg2(:,1)' ;
          eeg3(:,1)' ;
          eeg4(:,1)' ;
          eeg5(:,1)' ;
          eeg6(:,1)' ;
          eeg7(:,1)' ;
          eeg8(:,1)' ;
          eeg9(:,1)' ;
          eeg10(:,1)' ];
n = 10; %count data
a = [] ; %error matrix of all data
b = [] ; %mean error from a (trade off curve)
N = 10240; % sample of data in 20 second
BIC = [] ;
for p = 1:20 % loop for lag order 0 - 20
    Y = [] ;
    Y = data(:,p+1:N);

    X = [] ; % this loop
    for i = 1:N-p
        X1 = [1];
        for j = 1:p
            X1 = [X1 ; data(:,p+i-j)];
        end
        X = [X X1];
    end

    thetahat = Y*X'*inv(X*X');

    sigmahat = [0];
    X = [] ;
    for i = 1:N-p
        X1 = [1];
        for j = 1:p
            X1 = [X1 ; data(:,p+i-j)];
        end
        sigmahat = sigmahat + (data(:,p+1) - thetahat*X1)*(data(:,p+1) - thetahat*X1)';
    end
```

```

sigmahat = sigmahat/N-p;

% this is estimated EEG signal
Yhat = [data(:,1:p) thetahat*X];

/% find trade off curve
for k = 1:n
    error = 0;
    error = sum(abs(data(k,:)-Yhat(k,:)))/N;
    a(k,p) = error;
end

% mean error at p order, when finish p loop, b is mean error of order 0-20.
b(1,p) = sum(a(:,p))/n;

% find BIC score of p order 0-20
logLH = 0; % find log-likelihood function at p order
rlogLH = 0;
llogLH = (N-p)*log(det(inv(sigmahat)) ) /2;
X = [];
for i = 1:N-p
    X1 = [1];
    for j = 1:p
        X1 = [X1 ; data(:,p+i-j)];
    end
    rlogLH = rlogLH + (data(:,p+1) - thetahat*X1)'*inv(sigmahat)*(data(:,p+1) - thetahat*X1);
end

rlogLH = (-1/2)*rlogLH;
logLH = llogLH+rlogLH; %Log-likelihood function
d = n*(1+n*p);
BIC(1,p) = -2*logLH+ d*log(n);

end

% trade off curve
figure
plot(1:p,b(1,:),'-xb')
title('error of pth lag order')
xlabel('p order')
ylabel('error')

% BIC score
figure
plot(1:p,BIC(1,:),'-xb')
title('BIC score of pth lag order')
xlabel('p order')
ylabel('BIC')

% Wald test for LS problem when varying variance of noise
SX=100;
x=rand(SX,1)*10; % The ture value
for i=1:SX
    if rem(i,2)==0
        x(i)=0;
    end
end

```

```

A=rand(150,SX)*10;
mu=zeros(150,1);
alpha=0.05; c=chi2inv(1-alpha,1);
q=0;
Type1Error=zeros(1,20); Type2Error=zeros(1,20); n=1;
for i=0.5:0.5:10
    E1=0; E2=0; j=0;
    while j<10000
        S=i*norm(x); Sigma=S*eye(150);
        V=S*inv(A'*A);
        v=mvnrnd(mu,Sigma,1);
        y=A*x+v';
        clearvars v
        xhat=A\y;
        for m=1:SX
            R=zeros(1,SX);
            R(m)=1;
            J=R;
            r=R*xhat-q;
            W=r'*inv(J*V*J')*r;
            if W>c
                if rem(m,2)==0
                    E1=E1+1;
                end
            else if W<c
                if rem(m,2)==1
                    E2=E2+1;
                end
            end
        end
        j=j+1;
    end
    Type1Error(n)=E1/(10000*SX/2);
    Type2Error(n)=E2/(10000*SX/2);
    n=n+1;
end
i=(0.5:0.5:10);
Type1Error
Type2Error
plot(i,Type2Error,'-o'); ylabel('Type II Error'); xlabel('Multiplier');
title('Type II error when varying variance of noise');

% Wald test for LS problem when varying alpha
SX=100;
x=rand(SX,1)*10; % The ture value
for i=1:SX
    if rem(i,2)==0
        x(i)=0;
    end
end
A=rand(150,SX)*10;
mu=zeros(150,1);
S=10*norm(x);
Sigma=S*eye(150);
q=0;
V=S*inv(A'*A);
Type1Error=zeros(1,10); Type2Error=zeros(1,10); n=1;
for i=0.01:0.01:0.1

```

```

E1=0; E2=0; j=0;
while j<10000
    alpha=i; c=chi2inv(1-alpha,1);
    v=mvnrnd(mu,Sigma,1);
    y=A*x+v';
    clearvars v
    xhat=A\y;
    for m=1:SX
        R=zeros(1,SX);
        R(m)=1;
        J=R;
        r=R*xhat-q;
        W=r'*inv(J*V*J')*r;
        if W>c
            if rem(m,2)==0
                E1=E1+1;
            end
        else if W<c
            if rem(m,2)==1
                E2=E2+1;
            end
        end
    end
    j=j+1;
end
Type1Error(n)=E1/(10000*SX/2);
Type2Error(n)=E2/(10000*SX/2);
n=n+1;
end
i=0.01:0.01:0.1;
Type1Error
Type2Error
plot(i,Type2Error,'-o'); xlabel('Significance level'); ylabel('Type II Error')
title('Type II error when varying a significance level');

% Wald test for LS problem when varying a sample size
SX=100;
x=rand(SX,1)*10; % The ture value
for i=1:SX
    if rem(i,2)==0
        x(i)=0;
    end
end
A=rand(150,SX)*10;
mu=zeros(150,1);
S=10*norm(x);
alpha=0.05; c=chi2inv(1-alpha,1);
q=0;
Type1Error=zeros(1,50); Type2Error=zeros(1,50); n=1;
for i=101:150
    E1=0; E2=0; j=0;
    while j<10000
        A1=A(1:i,:);
        V=S*inv(A1'*A1);
        mu=zeros(i,1);
        Sigma=S*eye(i);
        v=mvnrnd(mu,Sigma,1);
        y=A1*x+v';

```

```

clearvars v
xhat=A1\y;
for m=1:SX
    R=zeros(1,SX);
    R(m)=1;
    J=R;
    r=R*xhat-q;
    W=r'*inv(J*V*J')*r;
    if W>c
        if rem(m,2)==0
            E1=E1+1;
        end
    else if W<c
        if rem(m,2)==1
            E2=E2+1;
        end
    end
    end
    j=j+1;
end
Type1Error(n)=E1/(10000*SX/2);
Type2Error(n)=E2/(10000*SX/2);
n=n+1;
end
i=101:150;
Type1Error
Type2Error
plot(i,Type2Error); ylabel('Type II Error'); xlabel('Sample size');
title('Type II error when varying a sample size');

% Wald test for AR model

%GENAR(n, p, N, d) generates order-p AR model sparse parameter Ai, i = 1,
%2, ..., p, with density d and simulates N data vectors from the generated
%model.
%Initial p data vectors are randomly generated and the data is corrupted by
%noise of variance varn.
%Assume that all the model parameters Ai, i = 1, 2, ..., p, share the same
%sparsity pattern.

function [A,TrueModel,Model] = ARWaldTest(n, p, N, d, varn,alpha)

Y = zeros(n, N-p);
A = zeros(n, n, p);
H = zeros(n*p, N-p);

%create sparsity pattern with nonzero diagonal elements
sppat = sprandn(n, n, d) | speye(n);

sdA = sqrt(0.01);      %variance of normally distributed parameter

%generate parameters
for i = 1:p
    A(:,:,i) = sdA*sprandn(sppat);
end

%check model stability
comp = [speye(n*(p-1)), sparse(n*(p-1), n)]; %complementary

```

```

ITER = 0;
while (abs(eigs([A(:, :); comp], 1)) >= 1) && (ITER<1000)
    for i = 1:p
        A(:, :, i) = sdA*sprandn(sppat);
    end
    ITER = ITER + 1;
end
if ITER == 1000
    fprintf('ERROR\n')
    return
end

%generate data
H(:, 1) = randn(n*p, 1);      %initialize p data vectors

sdn = sqrt(varn);
for j = 2:(N-p)
    ynext = A(:, :)*H(:, j-1) + sdn.*randn(n, 1);
    H(:, j) = [ynext; H(1:n*(p-1), j-1)];
end
Y = [H(1:n, 2:(N-p)), ynext];
%--- compute Sigmahat ---
Sigma=0;
Ahat=Y*H'/(H*H');
for t=1:N-p
    Sigma=Sigma+(Y(:,t)-Ahat*H(:,t))*(Y(:,t)-Ahat*H(:,t))';
end
Sigmahat=Sigma/(N-p);
%--- Generate new parameter for Wald test ---
y=zeros(n,1,N);
for i=1:N
    if i<p+1
        y(:, :, i)=H(n*(p-1)+1:n*p, i);
    elseif i>=p+1
        y(:, :, i)=Y(:, i-p);
    end
end
yy=[];
for i=1:N
    yy=[yy y(:, :, N-i+1)];
end
M=zeros(n,n*n*p,N-p);
for i=p+1:N
    yyy=[];
    for j=1:n
        yyy=[yyy yy(j,N-i+2:N-i+1+p)];
    end
    for k=1:n
        M(k, :, i-p)=[zeros(1,(k-1)*n*p) yyy zeros(1,n*n*p -n*p-(k-1)*n*p)];
    end
end
sum=0;
for i=1:N-p
    sum=sum+M(:, :, i)'*inv(Sigmahat)*M(:, :, i);
end
Avar=inv(sum);
Ahatt=zeros(n,n,p);
for i=1:p

```

```

Ahatt(:,:,i)=Ahat(1:n,n*(i-1)+1:n*i);
end
theta=zeros(n*n*p,1);
num=1;
for i=1:n
    for j=1:n
        for k=1:p
            theta(num)=Ahatt(i,j,k);
            num=num+1;
        end
    end
end
%--- Wald test ---;
c=chi2inv(1-alpha,p)
model=zeros(1,n*n);
for i=1:n*n
    D=[zeros(p,p*(i-1)) eye(p) zeros(p,n*n*p-p*(i-1)-p)];
    r=D*theta;
    W=r'/(D*Avar*D')*r;
    if W>c
        model(i)=1;
    else model(i)=0;
    end
end
Model=zeros(n,n);
for i=1:n
    Model(i,:)=model(n*(i-1)+1:n*i);
end
TrueModel=A(:,:,1);
for i=1:n
    for j=1:n
        if TrueModel(i,j)^=0
            TrueModel(i,j)=1;
        end
    end
end

```